Ansible AWX Tokens

1- Creating a new OAuth2 access token

Login to the AWX container then run the following command.

```
$ awx-manage create_oauth2_token --user ${userid}
```

This command will generate a new token for a specified user:

```
root@vps290950:~# docker exec -it awx_task /bin/bash
bash-4.4# awx-manage create_oauth2_token --user admin
T66PGVTXQ83Paw0aOlYPddPmAeQM2y
```

Using the API:

The API accessible via http://<awx_server>/api/ can be also used to create an OAuth2 access token.

```
curl -H "Authorization: Bearer <existing oauth2 access token>"\
     -H http://<awx_server>/api/<version>/tokens/ \
     -H "Content-Type: Application/json" -d @payload.json
```

payload.json content**:**

```
{
    "description": "",
    "application": null,
    "scope": "write"
}
```

```
$ curl -H "Authorization: Bearer T66PGVTXQ83Paw0aOlYPddPmAeQM2y"      http://51.254.116.97:8052/api/v2/to
kens/      -H "Content-Type: Application/json" -d @payload.json
{"id":4,"type":"o_auth2_access_token","url":"/api/v2/tokens/4/","related":{"user":"/api/v2/users/1/","act
ivity_stream":"/api/v2/tokens/4/activity_stream/"},"summary_fields":{"user":{"id":1,"username":"admin","f
irst_name":"","last_name":""}},"created":"2025-02-26T22:54:56.735814Z","modified":"2025-02-26T22:54:56.74
8836Z","description":"","user":1,"token":"bhynRspn7L2CclahidOACJvOiYGHoV","refresh_token":null,"applicati
on":null,"expires":"3024-06-29T22:54:56.730881Z","scope":"write"}
```

Using the GUI:

- Login to AWX with admin permissions
- Navigate to Users
- Select the username you wish to create a token for
- Click on tokens, then the green plus icon
- Application can be left empty, input a description and select a scope <r/w>

2- Revoking tokens

```
$awx-manage revoke_oauth2_tokens --user ${userid}
```

This command will revoke tokens assigned to a specified user.

```
bash-4.4# awx-manage revoke_oauth2_tokens --user admin
revoked OAuth2AccessToken 6X5TN7SphFMFio03mlwVIjLBtDDU7I
revoked OAuth2AccessToken TvTqyoxm6rq8S20dKpsFLwjx0E5RjM
revoked OAuth2AccessToken T66PGVTXQ83Paw0aOlYPddPmAeQM2y
revoked OAuth2AccessToken bhynRspn7L2CclahidOACJv0iYGHoV
```

3- OAuth2 Applications

If you are planning to create a client to connect to the AWS RESTful API, you will need to either first create an OAuth2 application or use an OAuth2 Token.
Each OAuth2 application is represented as an OAuth2 client on the AWX server.

One of the ways of creating an OAuth2 application is by invoking a HTTP POST to the following: **/api/v2/users/${userid}/applications** OR **api/v2/application**

The payload sent should conform to the following JSON.

JavaScript

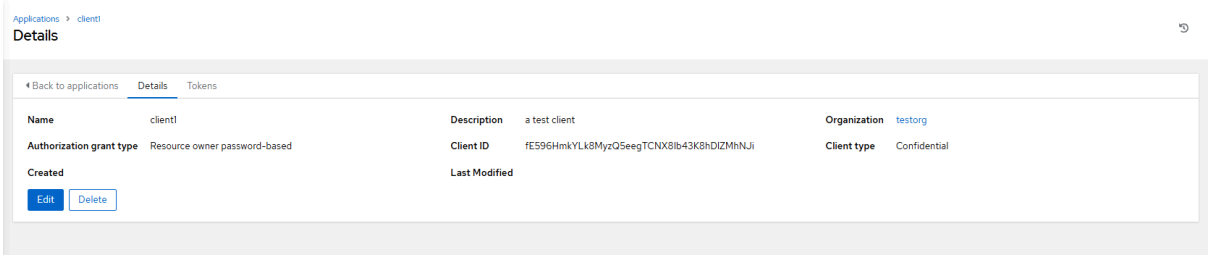
```
{
    "name": "client1",
    "description": "a test client",
    "client_type": "confidential",
    "redirect_uris": "",
    "authorization_grant_type": "password",
    "skip_authorization": false,
    "organization": 1
}
```

```
$ curl -H "Authorization: Bearer riSyB3keFNUQ6kftad9A11OI6Ol8fE"      http://51.254.116.97:8052/api/v2/ap
plications/      -H "Content-Type: Application/json" -d @payload.json
{"id":1,"type":"o_auth2_application","url":"/api/v2/applications/1/","related":{"named_url":"/api/v2/appl
ications/client1++testorg/","tokens":"/api/v2/applications/1/tokens/","activity_stream":"/api/v2/applicat
ions/1/activity_stream/"},"summary_fields":{"organization":{"id":1,"name":"testorg","description":""},"us
er_capabilities":{"edit":true,"delete":true},"tokens":{"count":0,"results":[]}},"created":"2025-02-26T23:
15:30.552461Z","modified":"2025-02-26T23:15:30.552575Z","name":"client1","description":"a test client","c
lient_id":"fE596HmkYLk8MyzQ5eegTCNX8Ib43K8hDlZMhNJi","client_secret":"Rk8kRcErnck2HzcWOIpj19p2XOll3Qcbmrt
Vm61YrhcsxUK6pEsT5kTVpNwJNAmPeO8vLsbqjqAgEeVHXFPum4h8BwwJREf3FeR3Q3sGehfAU2KkUtzOm7ORsoMUtKHA","client_ty
pe":"confidential","redirect_uris":"","authorization_grant_type":"password","skip_authorization":false,"o
rganization":1}
```

The generated response will be like the following:

```
Unset
{
    "id": 1,
    "type": "o_auth2_application",
    "url": "/api/v2/applications/1/",
    "related": {
        "named_url": "/api/v2/applications/client1++Default/",
        "tokens": "/api/v2/applications/1/tokens/",
        "activity_stream": "/api/v2/applications/1/activity_stream/"
    },
    "summary_fields": {
        "organization": {
            "id": 1,
            "name": "Default",
            "description": ""
        },
        "user_capabilities": {
            "edit": true,
            "delete": true
        },
        "tokens": {
            "count": 0,
            "results": []
        }
    },
    "created": "2025-02-02T10:04:40.303832Z",
    "modified": "2025-02-02T10:04:40.303929Z",
    "name": "client1",
    "description": "VBS test client",
    "client_id": "********************",
    "client_secret": "********************",
    "client_type": "confidential",
    "redirect_uris": "",
    "authorization_grant_type": "password",
    "skip_authorization": false,
    "organization": 1
}
```

The OAuth2 relevant fields are *client_id* and *client_secret*



4- Application access rules

Access rules for applications are as follows:

- System administrators can view and manipulate all applications in the system
- Organization administrators can view and manipulate all applications belonging to Organization members
- Other users can only view, update, and delete their own applications, but cannot create any new applications

Tokens, on the other hand, are resources used to actually authenticate incoming requests and mask the permissions of the underlying user.

There are two ways to create a token:

- **POST** to the **/api/v2/tokens/** endpoint with application and scope fields to point to the related application and specify token scope
- **POST** to the **/api/v2/applications/<pk>/tokens/** endpoint with the scope field (the parent application will be automatically linked).

We will use the **2nd** option and invoke a **HTTP POST** to endpoint **/api/v2/applications/1/tokens/** (this was taken from the earlier response when creating an OAuth2 application). This will return the following response.

```
Unset
{
    "id": 3,
    "type": "o_auth2_access_token",
    "url": "/api/v2/tokens/3/",
    "related": {
        "user": "/api/v2/users/1/",
        "application": "/api/v2/applications/1/",
        "activity_stream": "/api/v2/tokens/3/activity_stream/"
    },
    "summary_fields": {
        "user": {
            "id": 1,
            "username": "admin",
            "first_name": "",
            "last_name": ""
        },
        "application": {
            "id": 1,
            "name": "client1"
        }
    },
```

```
    "created": "2025-02-02T10:20:59.448195Z",
    "modified": "2025-02-02T10:21:00.563793Z",
    "description": "",
    "user": 1,
    "token": "123",
    "refresh_token": "teree",
    "application": 1,
    "expires": "3019-07-04T10:20:59.380688Z",
    "scope": "write"
}
```


4- Personal Access Tokens

A simple way to request an OAuth2 token. Invoke a HTTP POST to endpoint: `/api/v2/users/<userid>/personal_tokens/`

The payload should use the following format:

```
{
    "description":"Personal Tower CLI token",
    "application":null,
    "scope":"write"
}
```

5- OAuth2 Token expiry settings

Expiry settings for OAuth2 token's can be modified via the UI.

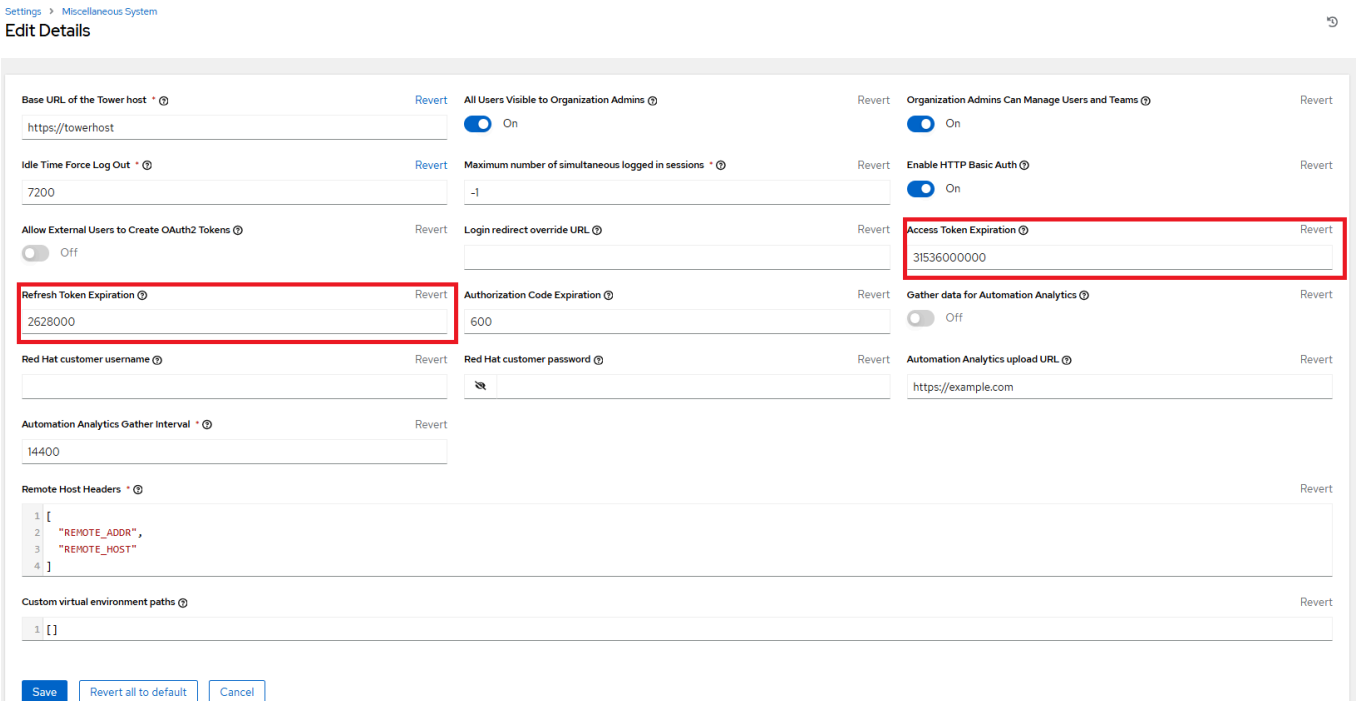
- Login to AWX as an admin user
- Click on **Settings** > **System**

- Refresh Token Expiration:

The duration (in seconds) refresh tokens remain valid after the expiration of their associated access token.

- Access Token Expiration:

The duration (in seconds) access tokens remain valid since their creation.


Settings > Miscellaneous System
Edit Details
Base URL of the Tower host * Revert
https://towerhost
All Users Visible to Organization Admins Revert
On
Organization Admins Can Manage Users and Teams Revert
On
Idle Time Force Log Out * Revert
7200
Maximum number of simultaneous logged in sessions * Revert
-1
Enable HTTP Basic Auth Revert
On
Allow External Users to Create OAuth2 Tokens Revert
Off
Login redirect override URL Revert
Access Token Expiration Revert
31536000000
Refresh Token Expiration Revert
2628000
Authorization Code Expiration Revert
600
Gather data for Automation Analytics Revert
Off
Red Hat customer username Revert
Red Hat customer password Revert
Automation Analytics upload URL Revert
https://example.com
Automation Analytics Gather Interval * Revert
14400
Remote Host Headers * Revert
[
"REMOTE_ADDR",
"REMOTE_HOST"
]
Custom virtual environment paths Revert
[]
Save
Revert all to default
Cancel